

Amendments to the Drawing:

As discussed above, all the tables have been moved into the drawing as FIGs. 12 - 68. These new FIGs. were made by simply electronically copying them from the original Specification and adding FIG. numbers. Thus, no new matter is presented.

Thus, FIGs. 12 - 68 are supplied herewith identified as "Supplemental Drawing Figures" in headers outside the sight line.

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 3.1.1.1 - FIG. 12

	GCOS Application	GFIP	UFIP	UNIX Application
1	application starts			
2	call X_ETL_OPEN	→		
3		send open request record		
4	←	return to application		
5			←	Call ETLAcceptOpen
6			receive open request record	
7			→	return to application

Table 3.1.1.2.1 - FIG. 13

	GCOS Application	GFIP	UFIP	UNIX Application
1	Application starts			
2	call X_ETL_OPEN call X_ETL_WRITEREC			call ETLAcceptOpen call ETLReadRecord
3	call X_ETL_CLOSE	→		
4		send close request record	→	
5			←	call ETLReadRecord
6			receive close request record	
7			return to application	EOF status
8			←	call ETLClose
9		←	send close response record	
10			return to application	→
11		receive close response record		
12	←	return to application		

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 3.1.1.2.2 - FIG. 14

	GCOS Application	GFIP	UFIP	UNIX Application
1	Application starts			
2	call X_ETL_OPEN call X_ETL_READREC			call ETLAcceptOpen call ETLWriteRecord
3				call ETLClose
4			send close request record	
5	call X_ETL_READREC			
6		receive close request record		
7	EOF status	return to application		
8	call ETLClose			
9		send close response record		
10		return to application		
11			receive close response record	
12			Return to application	

Table 3.1.1.3.1 - FIG. 15

	GCOS Application	GFIP	UFIP	UNIX Application
1	application starts			
2	call X_ETL_OPEN call X_ETL_WRITEREC			call ETLAcceptOpen call ETLReadRecord
3				call ETLClose
4			send error record	
5	call X_ETL_WRITEREC			
6		receive error record		
7		send error response record		
8	'etlconnerr' status	return to application	receive error response record	
9			return to application	

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 3.1.1.3.2 - FIG. 16

	GCOS Application	GFIP	UFIP	UNIX Application
1	application starts			
2	call X_ETL_OPEN call X_ETL_READREC			call ETLAcceptOpen call ETLWriteRecord
3	call X_ETL_CLOSE			
4		send error record	→	
5				call ETLWriteRecord
6			receive error record	
7		←	send error response record	
8		receive error response record		
9		return to application	return to application	'etlconnerr' status
10				call ETLClose
11			return to application	

Table 3.1.1.4.2 - FIG. 17B

	GCOS Application	GFIP	UFIP	UNIX Application
1	application starts			
2	call X_ETL_OPEN			call ETLAcceptOpen
3			←	call ETLWriteRecord
4			send data record	
5			return to application	→
6	call X_ETL_READREC	→		
7		receive data record		
8	←	return to application		
10				

Table 3.1.1.4.1 - FIG. 17A

	GCOS Application	GFIP	UFIP	UNIX Application
1	application starts			
2	call X_ETL_OPEN			call ETLAcceptOpen
3	call X_ETL_WRITEREC	→		
4		send data record		
5	←	return to application		
6			←	call ETLReadRecord
7			receive data record	
8			return to application	→

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 3.1.1.6.1 - FIG. 18

	GCOS Application	GFIP	UEXEC	UNIX Application
1	application starts			
2	call X_ETL_EXECUTE		Call ETLAcceptExcc	
3	call X_ETL_WRITEREC			
4		send execute record →		
5			receive execute record start UNIX application	application starts
6		send data record →		
7		return to application	receive data record write to 'stdin'	
8	Call X_ETL_WRITEREC			read stdin
9	call X_ETL_CLOSE			
10		send eof record →		
11		return to application	receive eof record close 'stdin'	
12				read stdin status = EOF
13				write stdout
14	call X_ETL_READREC		read stdout send data record ←	
15		read data record		
16		return to application		
17				close stdout exit ( )
18			receive EOF for stdout send eof record ←	
19	call X_ETL_READREC			
20		read eof record		
21	EOF status	return to application		
22	call X_ETL_TERMINATE			
23		send terminate record →		
24			read terminate record	
25			send terminate response exit ( ) ←	
26		read terminate response		
27		return to application		

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 3.1.1.7.1 - FIG. 19

	GCOS Application	GPIP	UFIP	UFAP
1	application starts			
2	call X_ETL_OPEN			
3	call X_ETL_WRITEREC			call ETLAcceptOpen call ETLReadRecord
4		send data record →		
5		return to application	receive data record	
6			return to application	
7				
8	Call X_ETL_CHECKPOINT			
9		send checkpoint record →		
10			receive checkpoint record	
11				write checkpoint file
12				Call ETLWriteRecord
13			← send checkpoint response	
14		receive checkpoint response		
15		return to application		

Table 3.1.1.7.2 - FIG. 20

	GCOS Application	GPIP	UFIP	UFAP
1	application starts			
2	call X_ETL_OPEN			
3	call X_ETL_WRITEREC			call ETLAcceptOpen call ETLReadRecord
4		send data record →		
5		return to application	receive data record	
6			return to application	
7				
8	Call X_ETL_ROLLBACK			
9		send rollback record →		
10			receive rollback record	
11				read checkpoint file perform rollback
12				Call ETLWriteRecord
13			← send rollback response	
14		receive rollback response		
15		return to application		

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 3.5.1.1 - Send Message that is the initial call to SID from a GCOS application: - FIG. 21

	Application	SID	IO_MAN	Application
1	application starts			
2	call send_message	→		
3		call activate_protocol_d river		
4		send link message		
5		send GCOS_stream message	→	
6			receive link message	
7			receive GCOS_stream message	
8			assign surrogate client process	→ application starts
9			send dbsp_link message	
10		receive dbsp_link message		
11		return to application		→ call receive_message

Table 3.5.1.2 - Send Message that is not the initial call to SID from a GCOS application: - FIG. 22

	Application	SID	IO_MAN	Application
1	call send_message	→		
2		send GCOS_stream message	→	
3		return to application	receive GCOS_stream message	
4				→ call receive_message

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 3.5.1.3 - Steady stream from a GCOS application (UNIX application is not receiving): - FIG. 23

	Application	SID	IO_MAN	Application
1	call send_message	→		
2		send GCOS_stream message	→	
3	←	return to application	receive GCOS_stream message	
4	repeat steps 1-3 multiple times			
5	call send_message	→		
6		SID exhausts credit from io_man		
7		SID issues a read I/O		
8		SID waits for a dbsp_credit message until a timeout occurs		
9	←	return timeout status		

Table 3.5.1.4 - Steady stream from a GCOS application (UNIX application is receiving): - FIG. 24

	Application	SID	IO_MAN	Application
1	call send_message	→		
2		send GCOS_stream message	→	
3	←	return to application	receive GCOS_stream message	
4				call receive_message
5	as steps 1-4 repeat	SID issues a read I/O		
6		←	send dbsp_credit message	
7		receive dbsp_credit message		



Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 3.5.2.1 - Receive Message that is the initial call to SID from a GCOS application: - FIG. 25

	Application	SID	IO_MAN	Application
1	application starts			
2	call receive_message	→		
3		call activate_protocol_driver		
4		send link message	→	
5			receive link message	
6			assign surrogate client process	→ application starts
7		←	send dbsp_link message	
8		receive dbsp_link message	←	call send_message
9		←	send dbsp_stream message	
10		receive dbsp_stream message	→ return to application	
11	←	return to application		

Table 3.5.2.2 - Receive Message that is not the initial call to SID from a GCOS application: - FIG. 26

	Application	SID	IO_MAN	Application
1	call receive_message	→		call send_message
2		←	send dbsp_stream message	
3		receive dbsp_stream message	→ return to application	
4	←	return to application		

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 3.5.2.3 - Steady stream to a GCOS application that is not calling Receive Message: - FIG. 27

	Application	SID	IO_MAN	Application
1				call send_message
2			send dbsp_stream message	
3		receive dbsp_stream message	return to application	
4				repeat steps 1-3 multiple times
5				call send_message
6			io_man exhausts credit from SID	
7			io_man waits for a GCOS_credit message until a timeout occurs	
8			return error status	

Table 3.5.2.4 - Steady stream to a GCOS application: - FIG. 28

	Application	SID	IO_MAN	Application
1				call send_message
2			send dbsp_stream message	
3		receive dbsp_stream message	return to application	
4	call receive_message			
5	as steps 1-4 repeat	send GCOS_credit message		
6			receive GCOS_credit message	

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

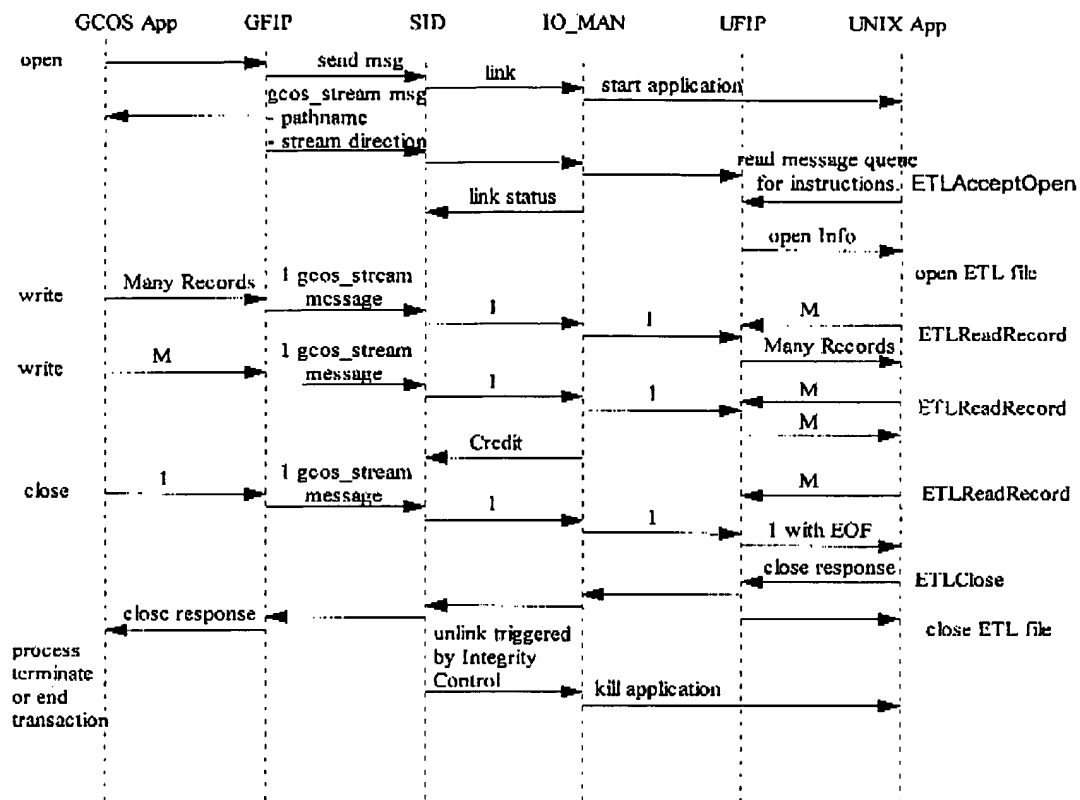


Table 3.5.2.5 - Component interactions for a GCOS stream - FIG.

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

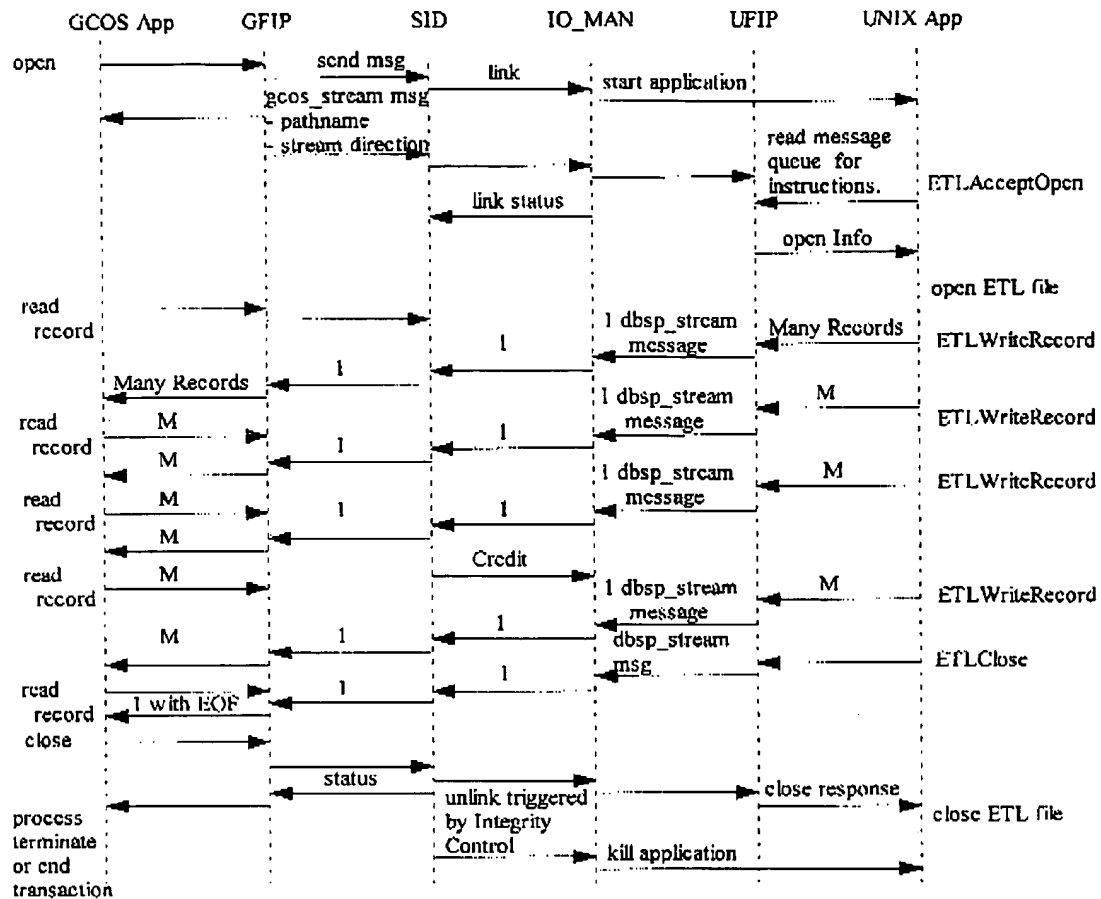


Table 3.5.2.6 - Component interactions for a UNIX stream - FIG.

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 3.6.3 - FIG. 31

variable name	description
maxHHInputSeqNum	This is the credit extended to GCOS, initialized to 8.
outMsgSeqNum	Controlled by IDUT, used to sequence output messages.
maxOutMsgSeqNum	This is the credit issued by GCOS, initialized to 8. This is checked by IDUT and if a dbsp_stream message has a msgSeqNum > maxOutMsgSeqNum, IDUT will return badCredit status to the requesting SC. Otherwise, the dbsp_stream message will be passed to GCOS and status of ok returned to the requesting SC. No other message type will be impeded by this.

Table 4.1.1 - FIG. 32

	0	3
		1
Word 0	socket header length in ASCII decimal = '0016'	
1	RFU	
2-3	length of the following message (buffer) in ASCII decimal (number of 8-bit bytes in the message plus the length of the Socket Header)	

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 4.1.2.1 - FIG. 33 GFIP/UFIP Buffer Format		
Bytes	Width	Field
00-03	04	ID = "ETL " (4 ASCII Characters)
04-07	04	Buffer_length (in 9-bit bytes)
08-11	04	Number of records
12-19	08	Buffer_sequence number
20-27	08	RFU
28-31	04	Debug_options
32-127	96	RFU
128-9999		Records

Table 4.1.3 - FIG. 34 Record Format		
Bytes	Width	Field
00-01	02	data size
02-03	02	record type
04-64003	<64004	Records

Table 4.1.4 - FIG. 35 Open Request Record Format		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-07	04	ETL-OPEN-ACTION
08-11	04	ETL-DATA-FORMAT
12-15	04	Length of path name
16-19	04	ETL-MAX-RESPONSE-TIME
20-23	04	ETL-RCV-REC-SIZE
24-63	40	RFU
64-127	64	UserID
128-xx	xx	Path Name (null terminated "C" string)

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 4.1.5 - FIG. 36		
Execute Request Record Format		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
05-08	04	Length of command
09-11	04	RFU
12-15	04	ETL-MAX-RESPONSE-TIME
16-64	48	RFU
65-127	64	UserID
128-xx		Command (null terminated "C" string)

Table 4.1.6 - FIG. 37		
4.1.5 Close Request Record Format		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-31	28	RFU

Table 4.1.7 - FIG. 38		
4.1.6 Close Response Record Format		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-31	28	RFU

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 4.1.8 - FIG. 39 Terminate Request Record Format		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-07	04	ETL-MAX-TIME
08-31	24	RFU

Table 4.1.9 - FIG. 40 Terminate Response		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-07	04	ETL-COMMAND-STATUS
08-31	24	RFU

Table 4.1.10 - FIG. 41 4.1.10 Data Record		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-xx	yy	Application Data

Table 4.1.11 - FIG. 42 Checkpoint Request Record Format		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-12	08	Last record number processed by GCOS
08-11	04	ETL-CKPT-DATA-LENGTH
12-31	20	RFU
32-xx	yy	ETL-CKPT-DATA



Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 4.1.11 - FIG. 42		
Checkpoint Request Record Format		
Bytes	Width	Field
Table 4.1.12 - FIG. 43		
Checkpoint Response Record Format		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-31	28	RFU

Table 4.1.13 - FIG. 44		
Rollback Request Record Format		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-31	28	RFU

Table 4.1.14 - FIG. 45		
Rollback Response Record Format		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-11	08	Last record number processed by GCOS
12-15	04	Checkpoint data length (bytes)
16-31	16	RFU
32-xx	Yy	Checkpoint data

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 4.1.15 - FIG. 46 Error Record Format		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-07	04	Error Source
08-11	04	Error Code
12-15	04	Length of Error Message
16-31	16	RFU
32-xx	yy	Error Message (null terminated "C" string)

Table 4.1.16 - FIG. 47 Error Response Record Format		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-31	28	RFU

Table 4.1.17 - FIG. 48 EOF Record Format		
Bytes	Width	Field
00-01	02	Data size
02-03	02	Record Type
04-31	28	RFU

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 4.3.1 - FIG. 49		
Restart Information File Format		
Bytes	Width	Field
00-03	04	ID = "ETL " (4 ASCII Characters)
04-07	04	Version
08-11	04	ETL-OPEN-ACTION
12-15	04	ETL-DATA-FORMAT
16-23	08	Number of data records processed since Open Request Record
20-23	04	Number of bytes in GCOS recovery data
24-27	04	ETL-CKPT-DATA-LENGTH
28-31	12	RFU
32-1055	1024	AIX path name
1056-xx	ww	ETL-CKPT-DATA

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 5.1.1 - FIG. 50

<u>Structure &amp; Name</u>	<u>Description</u>	<u>Default</u>
<u>Value</u>		
01 ETL-FILE-ID	COMP-6.	0
01 ETL-REC-LENGTH	COMP-6.	0
01 ETL-MAX-LENGTH	COMP-6.	0
01 ETL-RETURNED -LENGTH.	COMP-6.	0
01 ETL-PARAMETER-BLOCK.		
02 ETL-VERSION	COMP-6.	1
02 FILLER	PIC X(60).	
02 ETL-IPADDRESS	PIC X(256).	Spaces
02 ETL-PORT	COMP-6.	0
02 ETL-MAX-RESPONSE-TIME	COMP-6.	1000
02 ETL-OPEN-ACTION	COMP-6.	
88 ETLSEND		1
88 ETLRECV		2
02 ETL-DATA-FORMAT	COMP-6.	
88 ETLASCII		1
88 ETLRAW		2
88 ETLBITS		3
02 ETL-RCV-REC-SIZE	COMP-6	80.
02 FILLER	PIC X(128)	Low-
value		
01 ETL-CHECKPOINT-BLOCK.		
02 ETL-CKPT-DATA-LENGTH	COMP-6.	0
02 FILLER	PIC X(12).	
02 ETL-CKPT-DATA	PIC X(4096).	Low-
value		
01 ETL-STATUS-RTN.		
02 ETL-PRIMARY-STATUS	COMP-6.	
02 ETL-TOKEN	COMP-6.	
02 FILLER	COMP-6	
	OCCURS 2 TIMES.	
02 ETL-IMMEDIATE.		
03 ETL-IMMD-FCGI	PIC X(4).	
03 ETL-IMMD-FUNC	COMP-6.	
03 ETL-IMMD-MAJOR	COMP-6.	
03 ETL-IMMD-MINOR	COMP-6.	
02 ETL-ORIGINAL.		
03 ETL-ORIG-FCGI	PIC X(4).	
03 ETL-ORIG-FUNC	COMP-6.	

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

03 ETL-ORIG-MAJOR	COMP-6.
03 ETL-ORIG-MINOR	COMP-6.
02 ETL-ERROR-MSG.	
03 ETL-ERROR-MESSAGE-LENGTH	COMP-6.
03 ETL-ERROR-MESSAGE-TEXT	PIC X(1024).

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 5.1.2 - FIG. 51				
GCOS Status Return Structure and Codes				
Primary Status Codes (numerical value)	Major Status Codes (numerical value)	Minor Status Codes	Minor Code Value	Meaning
ETLSUCCESSFUL (0)	ETLSUCCESSFUL (0)	ETLSUCCESSFUL	0	Successful
ETLEXCEPTION (2)	ETLCOMPLETED (0)	ETLEOF	1	End of File
		ETLFORCED	2	Terminate forced due to timeout
		ETLSTDOUT	3	Stdout data discarded
		ETLFORCEDOUT	4	Forced termination and stdout discarded
		ETLBUFSIZE	5	Buffer too small for record
ETLSVCFAILED (3)	ETLUSERERR (1)	ETLINVID	1	ETL-FILE-ID is not valid.
		ETLINVACTION	2	ETL-OPEN-ACTION is not valid
		ETLINVFORMAT	3	ETL-DATA-FORMAT is not valid
		ETLINVRECLN	5	Record is too large
		ETLINVRCVSIZE	6	ETL-RCV-REC-SIZE is not valid
		ETLINVPATH	7	ETL-PATHNAME is all spaces
		ETLINVIPA	8	ETL-IPADDRESS is all spaces
		ETLSTATE	10	Call not valid for the connection state
		ETLARGMISSING	11	Too few arguments for the function
		ETLARGEXTRA	12	Too many arguments for the function
		ETLINVREC	13	Bit 0 of the GCOS 8 byte is not zero
	ETLSVCERR (2)	ETLCONNERR	1	Error from server
		ETLTIMEOUT	2	Maximum time has expired
		ETLSOCKERR	3	Error from socket routine
		ETLNOCONN	4	No connections available - all in use
	ETLINTERR (3)	ETLMODE	1	Invalid mode (LCB) detected
		ETLUBUFID	2	UNIX buffer - ID invalid
		ETLUBUFLEN	3	UNIX buffer - length invalid
		ETLUBUFCNT	4	UNIX buffer - record count invalid
		ETLUBUFREC	5	UNIX buffer - record length invalid
		ETLUBUFEMPTY	6	UNIX buffer - no records
		ETLSOCKHDR	7	Invalid socket header
		ETLSOCKHDRLEN	8	Invalid socket header length
		ETLUBUFSIZE	9	UNIX buffer larger than internal buffer
		ETLWRONGREC	10	Record invalid for connection state
		ETLERRCODE	11	Undefined error code in an error record

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 5.1.2 - FIG. 51				
GCOS Status Return Structure and Codes				
Primary Status Codes (numerical value)	Major Status Codes (numerical value)	Minor Status Codes	Minor Code Value	Meaning
		ETLERRMSGLEN	12	Invalid error message length

Table 5.1.3 - FIG. 52				
Status codes returned by X_ETL_OPEN				
Primary Status	Major Status	Minor Status	Minor Code Value	Meaning
ETLSUCCESSFUL	ETLSUCCESSFUL	ETLSUCCESSFUL	0	Successful
ETLSVCFAILED	ETLUSERERR	ETLINV ACTION	2	ETL-OPEN-ACTION is not valid
		ETLINVFORMAT	3	ETL-DATA-FORMAT is not valid
		ETLINVRCVSIZE	6	ETL-RCV-REC-SIZE is not valid
		ETLINVPATH	7	ETL-PATHNAME is all spaces
		ETLINVIPA	8	ETL-IPADDRESS is all spaces
		ETLARGMISSING	11	Too few arguments for the function
		ETLARGEXTRA	12	Too many arguments for the function
	ETLSVCERR	ETLSOCKERR	3	Error from socket routine
		ETLNOCONN	4	No connections available - all in use

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 5.1.4 - FIG. 53				
Status codes returned by X_ETL_EXECUTE				
Primary Status	Major Status	Minor Status	Minor Code Value	Meaning
ETLSUCCESSFUL	ETLSUCCESSFUL	ETLSUCCESSFUL	0	Successful
ETLSVCFAILED	ETLUSERERR	ETLINVIPA	8	ETL-IPADDRESS is all spaces
		ETLARGMISSING	11	Too few arguments for the function
		ETLARGEXTRA	12	Too many arguments for the function
	ETLSVCERR	ETLSOCKERR	3	Error from socket routine
		ETLNOCONN	4	No connections available - all in use

Table 5.1.5 - FIG. 54				
Status codes returned by X_ETL_CLOSE				
Primary Status	Major Status	Minor Status	Minor Code Value	Meaning
ETLSUCCESSFUL	ETLSUCCESSFUL	ETLSUCCESSFUL	0	Successful
ETLSVCFAILED	ETLUSERERR	ETLINVID	1	ETL-FILE-ID is not valid
		ETLSTATE	10	Call not valid for the connection state
		ETLARGMISSING	11	Too few arguments for the function
		ETLARGEXTRA	12	Too many arguments for the function
	ETLSVCERR	ETLCONNERR	1	Connection was aborted by the server
		ETLTIMEOUT	2	Maximum time has expired
		ETLSOCKERR	3	Error from socket routine



Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 5.1.6 - FIG. 55				
Status codes returned by X_ETL_TERMINATE				
Primary Status	Major Status	Minor Status	Minor Code Value	Meaning
ETLSUCCESSFUL	ETLSUCCESSFUL	ETLSUCCESSFUL	0	Successful
ETLEXCEPTION	ETLCOMPLETED	ETLFORCED	2	Terminate forced due to timeout
		ETLSTDOUT	3	Stdout data discarded
		ETLFORCEDOUT	4	Forced termination and stdout discarded
ETLSVCFAILED	ETLUSERERR	ETLINVID	1	ETL-FILE-ID is not valid
		ETLSTATE	10	Call not valid for the connection state
		ETLARGMISSING	11	Too few arguments for the function
		ETLARGEXTRA	12	Too many arguments for the function
	ETLSVCERR	ETLCONNERR	1	Connection was aborted by the server
		ETLTIMEOUT	2	Maximum time has expired
		ETLSOCKERR	3	Error from socket routine

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 5.1.7 - FIG. 56				
Status codes returned by X_ETL_WRITEREC				
Primary Status	Major Status	Minor Status	Minor Code Value	Meaning
ETLSUCCESSFUL	ETLSUCCESSFUL	ETLSUCCESSFUL	0	Successful
ETLSVCFAILED	ETLUSERERR	ETLINVID	1	ETL-FILE-ID is not valid
		ETLINVRECLN	5	Record is too large
		ETLSTATE	10	Call not valid for the connection state
		ETLARGMISSING	11	Too few arguments for the function
		ETLARGEXTRA	12	Too many arguments for the function
		ETLINVREC	13	Bit 0 of the GCOS 8 byte is not zero
	ETLSVCERR	ETLCONNERR	1	Connection was aborted by the server
		ETLTIMEOUT	2	Maximum time has expired
		ETLSOCKERR	3	Error from socket routine

Table 5.1.8 - FIG. 57				
Status codes returned by X_ETL_READREC				
Primary Status	Major Status	Minor Status	Minor Code Value	Meaning
ETLSUCCESSFUL	ETLSUCCESSFUL	ETLSUCCESSFUL	0	Successful
ETLEXCEPTION	ETLCOMPLETED	ETLEOF	1	End of File
		ETLBUFSIZE	5	Buffer too small for record
ETLSVCFAILED	ETLUSERERR	ETLINVID	1	ETL-FILE-ID is not valid
		ETLSTATE	10	Call not valid for the connection state
		ETLARGMISSING	11	Too few arguments for the function
		ETLARGEXTRA	12	Too many arguments for the function
	ETLSVCERR	ETLCONNERR	1	Connection was aborted by the server
		ETLTIMEOUT	2	Maximum time has expired
		ETLSOCKERR	3	Error from socket routine

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 5.1.9 - FIG. 58				
Status codes returned by X_ETL_CHECKPOINT				
Primary Status	Major Status	Minor Status	Minor Code Value	Meaning
ETLSUCCESSFUL	ETLSUCCESSFUL	ETLSUCCESSFUL	0	Successful
ETLSVCFAILED	ETLUSERERR	ETLINVID	1	ETL-FILE-ID is not valid
		ETLSTATE	10	Call not valid for the connection state
		ETLARGMISSING	11	Too few arguments for the function
		ETLARGEXTRA	12	Too many arguments for the function
	ETLSVCERR	ETLCONNERR	1	Connection was aborted by the server
		ETLTIMEOUT	2	Maximum time has expired
		ETLSOCKERR	3	Error from socket routine

Table 5.1.10 - FIG. 59				
Status codes returned by X_ETL_RESTART				
Primary Status	Major Status	Minor Status	Minor Code Value	Meaning
ETLSUCCESSFUL	ETLSUCCESSFUL	ETLSUCCESSFUL	0	Successful
ETLSVCFAILED	ETLUSERERR	ETLINVID	1	ETL-FILE-ID is not valid
		ETLSTATE	10	Call not valid for the connection state
		ETLARGMISSING	11	Too few arguments for the function
		ETLARGEXTRA	12	Too many arguments for the function
	ETLSVCERR	ETLCONNERR	1	Connection was aborted by the server
		ETLTIMEOUT	2	Maximum time has expired
		ETLSOCKERR	3	Error from socket routine

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 5.1.11 - FIG. 60				
Status codes returned by X_ETL_DEFINEREC				
Primary Status	Major Status	Minor Status	Minor Code Value	Meaning
ETLSUCCESSFUL	ETLSUCCESSFUL	ETLSUCCESSFUL	0	Successful
ETLSVCFAILED	ETLUSERERR	ETLINVID	1	ETL-FILE-ID is not valid
		ETLINVRECLN	5	Record is too large
		ETLSTATE	10	Call not valid for the connection state
		ETLARGMISSING	11	Too few arguments for the function
		ETLARGEXTRA	12	Too many arguments for the function

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 5.1.12.1 - FIG. 61A State Transitions for connections established by X_ETL_OPEN				
ETL Functions	Connection States			
	1 RESET	2 WRIT- ING	3 READ- ING	4 CLOS- ING
OPEN (etlwrite)	2	/	/	/
OPEN (etlread)	3	/	/	/
OPEN [error]	=	/	/	/
WRITE	X	=	X	X
WRITE [fatal error]	X	4	X	X
READ	X	X	=	X
READ [fatal error]	X	X	4	X
READ [eof]	X	X	4	X
CHECKPOINT	X	=	=	X
CHECKPOINT [fatal error]	X	4	4	X
ROLLBACK	X	=	=	2/3 <sup>1</sup>
ROLLBACK [fatal error]	X	4	4	4
CLOSE	X	1	1	1
CLOSE [fatal error]	X	1	1	1

<sup>1</sup> The destination state is 'writing' (state 2) if ETL-OPEN-ACTION was ETLWRITE, and the destination state is 'reading' (state 3) if ETL-OPEN-ACTION was ETLREAD.

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 5.1.12.2 - FIG. 61B State Transitions for connections established by X_ETL_EXECUTE					
ETL Functions	Connection States				
	1 RESET	5 EXECUTING	6 STDIN CLOSED	7 STDOUT CLOSED	8 ENDING
EXECUTE	5	/	/	/	/
EXECUTE [error]	=	/	/	/	/
WRITE	X	=	X	=	X
WRITE [fatal error]	X	8	X	8	X
READ	X	=	=	X	X
READ [fatal error]	X	8	8	X	X
READ [eof]	X	7	8	X	X
CHECKPOINT	X	=	=	=	X
CHECKPOINT [fatal error]	X	8	8	8	X
ROLLBACK	X	=	5 <sup>2</sup>	5 <sup>3</sup>	5 <sup>4</sup>
ROLLBACK [fatal error]	X	8	8	8	8
CLOSE	X	6	X	8	X
CLOSE [fatal error]	X	8	X	8	X
TERMINATE	X	1	1	1	1
TERMINATE [fatal error]	X	1	1	1	1

Table 5.2.1 - Prefix characters - FIG. 62

Prefix values received by UNIX	Prefix values sent by UNIX on stdout
'*ETL*C' - checkpoint request record	'*ETL*C' - checkpoint response record
'*ETL*R' - rollback request record	'*ETL*R' - rollback response record

<sup>2</sup> The destination state is 'executing' (state 5) even though the last successful checkpoint may have been performed in the 'stdin closed' state (state 6).

<sup>3</sup> The destination state is 'executing' (state 5) even though the last successful checkpoint may have been performed in the 'stdout closed' state (state 7).

<sup>4</sup> The destination state is 'executing' (state 5) even though the last successful checkpoint may have been performed in the 'stdin closed' state (state 6) or the 'stdout closed' state (state 7).

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 6.1.1 - FIG. 63

	0	1 1 7 8	3 5
Word 0	Version Number		
1	Function Code		
2	Output Buffer Size		
3-5	RFU		
6	Maximum Delay Time		

Table 6.1.2 - FIG. 64

	0	11 78	3 5	
Word 0	FCGI			Immediate Status
1	Sub-Function	Major Status	Minor Status	
2	FCGI			Original Status
3	Sub-Function	Major Status	Minor Status	
4-6	RFU			
7	Error Message Line Length	Number of Error Message Lines		
8-85	Error Message Area - (78 words)			

Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 6.1.3.1 - FIG. 65

	0	1 1 7 8	3 5
Word 0-2		DBSS Name	
3-10		Application Name	
11-18		Executable Name	
19		Commitment Mode	
20-21		Command Name	
22		HA Preference	
23		File ID	
24-31		RFU	

Table 6.1.3.2 - FIG. 66

Value	Meaning
3	Oracle mode (no commitment coordination)

Table 6.1.3.3 - FIG. 67

Value	Meaning
0	no preference
1	assign the application to HA1
2	assign the application to HA2



Supplemental Drawing Figures - 09/896,700  
In Response to Office Action Mailed April 6, 2004+

Table 6.2.1.1 - FIG. 68

	0	1 1 7 8	3 5
Word 0		Version Number	
1		Function Code	
2		RFU	
3		Input Buffer Size	
4-5		RFU	
6		Maximum Delay Time	